

Retrospectiva del primer Sprint

Grupo G41

Alejandro Rodríguez López

Manuel Revilla Hernández

Juan Francisco Mier Montoto

Rubén Martínez Ginzo

Estrella de mar



Keep Doing: Peer reviews, reuniones, comunicación

Consideramos que las frecuentes reuniones, revisiones de código y en general comunicación entre todos los miembros del equipo han sido, son y serán factores clave en el desarrollo satisfactorio del proyecto. Aunque cada persona no conoce absolutamente todas las esquinas del código, cada uno conoce la estructura general del código, el estado del proyecto en cada momento, en qué se está trabajando y quién trabaja en qué parte.

Sin duda, la comunicación ha sido totalmente vital y el objetivo es mantenerla, añadiendo además que ahora ya estamos a programar en grupo y tenemos una mentalidad más ajustada a lo que se nos pide en este proyecto.

More Of: Código reutilizable

Uno de los grandes problemas que hemos detectado durante las reuniones y comunicaciones es la falta de reutilización de código dentro del propio proyecto para funciones con objetivos iguales o similares.

Si conseguimos reestructurar el código y reutilizar más partes del código, podemos reutilizar mucho del esfuerzo que se está empleando actualmente en mejorar otras partes del código, realizar más pruebas o realizar documentación, por ejemplo.

Start Doing: Documentación, pruebas y tests, lógica de negocio en el Model

- *Sobre la documentación:* hasta este primer sprint no hemos desarrollado ningún tipo de documentación acerca del proyecto, por lo que es ahora mismo uno de los objetivos más importantes para el segundo. Se necesitan más prototipos de interfaces, documentación en JavaDoc y explicaciones más abstractas sobre el código a nivel de proyecto.
- *Sobre pruebas y tests:* durante la revisión del primer sprint se detectaron numerosos fallos que se podían haber evitado fácilmente mediante pruebas sencillas. Además, es hora de ir planeando el desarrollo de tests mediante JUnit. En resumen, se necesita probar más el código, incluyendo el flujo de información a lo largo de las diversas pantallas y el funcionamiento del programa en casos extremos.
- *Sobre la lógica de negocio:* como señalado en la práctica de aula, hay que separar mejor la lógica de negocio entre el *Controller* y el *Model*, siguiendo con la descripción del modelo que estamos utilizando. El *Controller* debería ser tan solo el enlace entre la vista y el modelo.

Stop Doing: Merges grandes y conflictivos

Puesto que este primer sprint ha servido como introducción al trabajo grupal en Git y Github, se han cometido numerosos errores de sincronización de trabajo, incluyendo merges gigantes con muchos conflictos, rebases incorrectos, pérdida de información, actualización innecesaria del *main*, entre otros.

Seguimos trabajando para mejorar nuestro flujo de trabajo y aprender a aprovechar estas herramientas de trabajo.

Less of: Kanban

Hasta ahora, el hecho de utilizar Kanban, mantenerlo actualizado y atender a las consultas del cliente/product owner/profesor ha sido más negativo que positivo.

- Hemos tenido problemas al interpretar y refinar las historias de usuario.
- Hemos perdido tiempo de trabajo esperando la verificación y ajuste de las mismas.
- La lenta respuesta de preguntas dejan en entredicho el desarrollo de grandes porciones de código que son vitales para el funcionamiento de historias de usuario completas.

Por ello, hemos decidido que el uso de las herramientas proporcionadas no es prioritario al desarrollo del resto del proyecto, motivo por el cuál no se tendrá en cuenta para siguientes sprints. De momento, el desarrollo de código continúa pese a la falta de comunicación con el cliente o la grave falta de verificación del *product owner*.